Project Update

Keith Ballard

To create a thermomechanical solver, I broke the problem up into several steps:

1. Create a solver for the heat equation supporting specified temperatures and fluxes
2. Create a solver for the isotropic elasticity equation supporting specified displacements and tractions
3. Couple the two solvers by passing the temperatures from the heat equation solver to the elasticity equation solver
4. Add support for transient analysis (i.e. introduce time dependency)
5. Parallelize the solvers for distributed systems

This is the same process I followed when creating a thermomechanical solver in my research groups tools (Beta); however, this process created several problems. The thermal solver had a DoF handler that had only scalars (dim = 1), whereas the elasticity solver had a DoF handler supporting a vector valued problem (dim = 2 or 3). The elasticity problem needed the gradients of the thermal solution but the elasticity solver’s DoF handler could not interpolate a scalar function since it is a vector valued DoF handler. This meant that the thermal solver needed to compute the gradients and pass the gradients to the elasticity solver rather than the temperatures. The gradients are computed for each quadrature point of each cell leading to a vector by cells of a vector by quadrature points of values, which leads to a hackish way of passing the data. Since both solvers use the same triangulation, I assumed that cell iterators would loop over the cells in the same order, so the vector by cells is simply follows the order of the iterator during the loop. Of course, this limits the solvers to not renumber the DoF’s; otherwise, the gradients would be applied to the wrong cells when the elasticity solver loops through them. This also forces both solvers to use the same quadrature scheme, which is fine for now since I cannot think of a reason I would need them to have different quadrature schemes.

Another problem I ran into with the different DoF handlers concerning output. I tried to pass the thermal solution to the elasticity solver to be output to the same VTK file as the displacements. However, I found that the same DoF handler problem persisted. I created another hackish way to visualize both the displacements and the temperatures on the same mesh by a very simple script that reads the thermal and elasticity VTK output files and creates a combined output file. This is post processing, so it is not what I intended in the design.

It was suggested by Dr. Bangerth to combine the solvers leading to single DoF handler using a block stiffness matrix and right hand side. This definitely solves both of the problems I have encountered, and allows two-way thermal coupling. Granted in my field, the thermal input into aeroelastic structures due to turbulent flow, the temperature differential between the flow and the structure, or several other drivers is much larger than the thermal energy released by deformation. On the other hand, predicting effective properties given a microstructure could be more accurate using two way coupling. So it really comes down to the application of the solvers. The only disadvantage of the single DoF handler is that it makes the solver more complicated since it needs to solve both the heat and elasticity equations. This leads to a much larger assembly method, but this is a very minor drawback that can be overcome with some modularity and encapsulation. Documentation will become more important as this continues.

At the moment, I have the one way coupling without time dependence using the method I have described. I have begun writing the single DoF handler version of the problem, but it is not complete. Clearly this is the direction the design should be driven, since the benefits greatly outweigh the disadvantages.